R E P O R T R E S U M E S

ED 018 107 EF 001 578

ARCHITECTURAL PROGRAMMING--STATE OF THE ART.

BY- EVANS, BENJAMIN H.
AMERICAN INSTITUTE OF ARCHITECTS, WASHINGTON, D.C.

PUB DATE OCT 67

EDRS PRICE MF-$0.25 HC-$0.76 17P.

DESCRIPTORS- *ARCHITECTURE, *ARCHITECTURAL PROGRAMING, *BUILDING DESIGN, *DESIGN NEEDS, *SYSTEMS ANALYSIS, BUILDING IMPROVEMENT, COMMUNICATION PROBLEMS, CONSTRUCTION COSTS, DECISION MAKING, SPECIALIZATION,

IN RESPONSE TO A NEED FOR A MORE THOROUGH AND RIGOROUS STUDY AND ANALYSIS PROCESS IN ENVIRONMENTAL FUNCTIONS PRIOR TO THE DESIGN OF NEW BUILDINGS, A STUDY WAS UNDERTAKEN TO IDENTIFY THE EMERGING TECHNIQUES OF ARCHITECTURAL PROGRAMING PRACTICE. THE STUDY INCLUDED CORRESPONDENCE AND REVIEW OF PERIODICALS, QUESTIONNAIRES AND VISITATIONS, AND A WORKSHOP OF PROGRAMING. TOPICS MENTIONED WERE--(1) PROBLEMS OF COMMUNICATION AND RESPONSIBILITY, (2) SOURCES OF PROGRAMS, (3) TECHNIQUES USED, AND (4) THE DEVELOPMENT OF SPECIALISTS. TWO EXAMPLES ARE GIVEN OF AN ARCHITECTURAL FIRM AND A GROUP OF PROGRAMING CONSULTANTS. OTHER ORGANIZATIONS AND APPROACHES ARE ALSO MENTIONED. SUGGESTIONS FOR RESEARCH AND DEVELOPMENT ARE INDICATED. THIS PAPER WAS PRESENTED AT THE AIA ARCHITECT-RESEARCHERS' CONFERENCE, GATLINBURG, TENNESSEE, OCTOBER 25, 1967. (MM)

October, 1967

"ARCHITECTURAL PROGRAMMING - STATE OF THE ART"

By Benjamin H. Evans, AIA
Director of Education and Research Programs
American Institute of Architects

It has been true in the past that we have frequently designed buildings without any clear concept of what was going to take place in those buildings. Architects and owners have too frequently made decisions without any real study of their present or future needs. Because they were usually dealing with historic building types, they knew somewhat intuitively what would suffice to meet the particular functions of that already well established pattern of activities. Everybody knows what's needed in a school.

Even with the present-day diversification of necessary functions, some completely new in the societal pattern, we still find we are able to adopt some old buildings and spaces to fit the new needs. That we are able to frequently do so, somewhat conceals the fact that in thousands of other cases, old buildings that are structurally sound are being razed because they have outlived their functional usefulness and cannot be adapted. It is not difficult at all to make a strong argument in favor of the need for a much more thorough and rigorous study and analysis process of environmental functions prior to the design of new buildings. I would venture to say that almost everyone in the design field would agree, in principle, on this point.

Before I go further, however, I should pause and define the term programming since I will be using it frequently and since I find so many different interpretations of the word. One of the most pressing problems in discussing this area of activity is the different uses to which the word programming is put.

When I say programming, I am referring to the process by which the criteria is developed for the design of a space, building, facility, or in broader terms an environment, or any piece of it. Such a program of criteria will include functional requirements, aesthetic requirements, "spiritual" requirements, and/or any other requirements that may influence the decision making. Such a program will usually result in a written report, but that report itself is not the program, but merely a "statement" of the program at some particular point in time. Programs are not generally definitive objects, but continuing processes which never really reach the absolute. They continue to change and recycle throughout the design process, sometimes through the construction process, and certainly through the life of the facility.

On the basis of my study and investigation, I have found very few people who are really trying to program in any kind of a systematic and analytic manner. Frankly, the big problem seems to be that they don't really know how, or how far in the direction of a serious study they should or want to go. There are, of course, many good programmers around and many examples of good programs, but such examples are generally not accomplished by a systematic process - a process which can be transferred to another programmer. They usually are the result of bright programmers who instinctively know the right questions to ask.

Sensing that a problem existed, the Committee on Research for Architecture of the AIA, directed me, with the help of the Committee and Professor Herbert Wheeler at Penn State University, to undertake a study of the "emerging techniques" of programming practice, particularly among architects, but also wherever I could find significant practices underway that would provide a valuable input into the bank of general knowledge on the subject.

By correspondence and a review of the periodicals, I attempted to seek out those people who were doing significant work in programming. It took some time, but I finally began to open the channels of communication. After collecting a list of names, I sent questionnaires to get further information and finally made a series of visitations. I suppose in the end I found that about one out of ten names that were given to me proved to be involved in a programming process which I felt had anything significant to offer the profession.

As a result of my contacts, and in cooperation with the Building Research Institute, I called a workshop of some 22 of the people, with whom I had come in contact, to meet in Washington for a general discussion of programming. The meeting was significant, I felt, because it was the first time that many of these people, all actively working in programming, had met each other or even heard of each other's work. The lack of communication in the field was obviously significant. As a result of that meeting, some six research papers on the subject of programming will be presented at a regular session of the BRI Conferences here in Washington in November.

Let me quickly review the highlights of what my survey has turned up, then I'll get into some of the specifics of what programmers are doing.

Major Problems

The major problem seems to be the matter of communications between client, user, programmer and designer. Again and again, I heard comments about the difficulties of clients understanding the programmer's objectives; about the programmer's inability to get at the user's requirements; about the designer's difficulties understanding the programmer; about the difficulty of communicating with the psychiatrist and other consultants; about the designer's difficulties in convincing the client that certain intangibles should be considered. It sometimes seems that weak designs are the result of inabilities to communicate. The architect feels that he has a great contribution to make by seeking out the intangibles, by finding out what kind of an "aesthetic and spiritual" environment can be developed that will provide for the client and his users beyond the simple mechanical necessities. To the architect, a building which just works is not enough - - it has to be a building which inspires man's most noble inspirations as well as providing for his shelter needs. To do so requires time and energy - both on the part of the designer and the client. It doesn't always result in a more costly building, but it nearly always results in a better building.

Nevertheless, the architect often has difficulty in communicating such ideas to his client and convincing his client that such human needs are worthy of special attention and require deeper probing at the programming stage.

Another major problem seems to be in the difficulty of getting down to the essence of the owner's intent or objective. What is the real problem to be solved? What are the hidden implications of significance beyond that which everybody easily recognizes? Getting at the essence of the problem seems too often to depend on intuition or just plain luck.

Among the other problems repeatedly cited were: serious programming is not done frequently enough; programmers do not have a sufficiently thorough knowledge of the client's operations; programs are often too rigid and inflexible, and; it is completed to see whether it met the program needs.

Responses to the questionnaire I sent indicate that almost all architects prefer to do the program themselves; so do the independent programmers, the institutional clients and everybody else. It seems that everyone involved is reluctant to turn the programming over to someone else. This appears to be a reflection of the old philosophy: "If you want it done right, do it yourself."

Most architects do not charge the client extra for programming even though this is not generally considered a part of their basic services and programming does cost them money - sometimes a significant amount. Clients would be wise to realize that there is no substitute for good planning. On some jobs, the design criteria are so simple as to be pretty well derived in a few days time in which case the cost is negligible. But in most cases, the client would profit from a thorough programming stage. Among the architects who keep records of their costs, sufficiently for determining how much of their fee they spend

on programming, the figure ranges from 1% to 10% with about 6% the average.

Almost all the persons interviewed said that they used a "systematic" programming process, but their answers to other questions would seem to leave some doubt. Few of them use any kind of forms for their interviewing except for very large jobs.

For private individual clients, only about 1 in 5 provides a program - 1 in 2 for corporate clients. On federal and state construction jobs, the architect is given a program about 5 out of six times, which certainly points up that our government agencies would seem to be better clients than the general public. By way of further explanation, however, another study in which I'm involved on the Cost of Architectural Services indicates that architects generally receive less fee and make less profit on government jobs. It sounds as if our federal friends may be smarter than we think!

On the whole, architects say that only about 50% of the programs they receive from other sources are rated as good. Generally, they feel they have to do the programs over or at least rewrite them. Analysis of the architects' financial records reveals that on the jobs where the programs are rated as bad, the architects tend to make less profit - for obvious reasons. Good programs clarify the job to be done - cut down on the time necessary in client checks, and on changes and revisions of drawings, and specifications.

Emerging Techniques

In spite of the general lack of a systematic approach to programming, my study shows that there are many emerging techniques, or sparks of

ideas, hither and yon that show inspiration and are worthy of note. Professor Herbert Wheeler sent me a list of items that he has uncovered in his part of the study. His list numbers over a hundred ideas which he gleaned from studying program statements from throughout the country. They include such things as:

Site analysis diagrams

Zoning and building code check lists

Desire line charts for transportation

Cartographaton prints of trip data

Room data forms

Performance criteria listing

Block space diagrams - 3 dimensional

Space-to-space relationship diagrams

Role relationships statements

Deconfuse users programs, and many others.

At this point in the study, I can't begin to define all of these ideas, but they are indicative of the creative thought being given to programming.

There appears to be a definite trend toward the development of specialists in programming - that is, persons who have studied and worked specifically in the program area. These people may work for architects, programming organizations, for corporate clients or for educational institutions. Most of them have backgrounds in architecture, but not all. A combination of architectural, business and management backgrounds appears to be especially suitable for the programming task.

Some architects feel that architects are the only ones who should do programming.Some would prefer not to be involved until the program has been completed. Some feel that program specialists are perfectly acceptable. All of these would agree that programs must be developed by competent people, and most would argue that the programmer must have an architectural background. The independent programmers argue that they are in a better position than the architect to program in a completely unbiased and systematic manner. The architects argue that the independent programmers have no understanding or feel for the creative aspects of programming. Probably all of these viewpoints have some validity.

However, the inescapable point is that programming takes hard work, study, talent, and intelligence and it may be found almost anywhere, but seldom as frequently as it is needed. It is also generally agreed that regardless of who does the programming, the architect-designer must be brought into the process before it has ended, and the programmer must continue to work with the designer even after the preliminary design phase.

Examples

There are several examples of what I believe to be well organized and somewhat unique activities in programming, and for the next few moments I'd like to describe some of these. I do not suggest that these are the best examples or that these firms are any more competent than others - merely that these have interesting characteristics.

In the office of Caudill, Rowlett and Scott, Architects and Planners in Houston, Texas, I found one of the most organized efforts of all

those I visited. The fact that the firm is a relatively large one makes it possible for them to go about their programming in a way that smaller firms cannot do, but nevertheless their efforts are significant and relevant to all architects.

The effort at CRS is directed by partner Willie Pena and has a relatively short "organized" history. Pena has described the CRS Programming Section as concerned with three phases of activity: (1) to do research and collect background information for use by the designers, (2) to teach their project managers how to program and to follow the firm's programming philosophy, and (3) to provide service to clients to assist them in their programming efforts. Pena believes that some day the programming section will work itself out of a job when all of their project managers are each fully prepared to handle their own programming efforts.

CRS believes in the problem solving approach - that architecture is a matter of solving problems and that good architecture usually depends on first having a good statement of what the problems are, or as Bill Caudill is fond of saying, "You can't solve the problem until you know what the problem is."

CRS also believes in the team concept and approaches problem solving as a group of professionals, each with different talents and skills - talents and skills to be brought in where needed to work as a team.

And thirdly, they believe in the "squatters" technique. "Squatters" is a term applied to the CRS technique of sending a team of professionals to an out of town site to actually develop the preliminary design on site. They usually set up office in a building close by and usually

in association with a local architect. It allows opportunity for the team to become personally acquainted with the client, to observe the site and the locale first hand, and to bring the client into the design study.

This overall philosophy is important in order to understand the place of programming in their operation.

The purpose of the programming section at CRS is to take the prime responsibility for programming out of the hands of the project managers and designers, and put it into the hands of experts in order to approach the planning systematically and objectively and to avoid inadvertantly influencing the client with a biased viewpoint. As you are probably well aware, designers sometimes involve their own prejudices and pet theories in design, to the point that the owner may wonder whose building the finished product really is.

Willie Pena believes that programming is first a systematic, analytic process and that it is not subjective. Secondly, that it must be creative, but it must not be blind to the facts. To quote Pena, "You first have to get at the essence of the problems before you get next to the client's heartbeat. In getting at the essence of the problem, you have to think creativily about what's unique to the situation; about what the big coneepts are."

Then, CRS finds that they can't really separate program from cost and site. These three things, program, cost and site are the principle ingredients which give the building its principle form - the big form givers. CRS looks at the program in terms of aims, methods and people.

They look at the site in terms of quality and design possibilities. They look at costs in terms of initial budget, operating costs, and long term costs. All of these will have a bearing on the final program.

At CRS the sequence of programming follows this pattern:

1) Establish the client's aims
2) Collect, organize and analyze the facts
3) Uncover and develop concepts
4) Establish needs, and
5) Develop the problem statement, or brief.

Pena says that one of the real problems in programming is communicating with the client. CRS places great emphasis on keeping the client "tuned in" which they do in part through a series of graphic techniques. They speak to the client with analysis cards (or snow cards) and with brown-paper sheets. The 5x7 analysis cards are used for describing each of the many points on which the client should be informed - one idea to each card for emphasis. The cards may deal with several specific points about the site which will influence the design or they may relate relevant climatic data. They may state code and zoning restrictions. The important thing is that the client look thoroughly and systematically at the factors which will influence the design of his building and understand why they are significant.

On brown-paper work sheets pinned to the wall, CRS illustrates for the client space quantities by drawing small squares. The various siged squares demonstrate th the client the relative quantities of space allocated to the various functions so that the client can see the implications of his earlier desisions and participate in the final decisions. Ac changes are called for, new brown sheets are pasted over the old ones

and new sketches drawn until the sheets are almost book thick.

The CRS programming effort is systematic, analytic and creative. One would like to think that the fine quality of their designs reflects a good programming foundation.

A different kind of a firm with which I was quite impressed was Becker and Becker Associates, planning and design consultants, in New York City. Nat Becker, the President of the organization, is an industrial designer, but his staff includes people from several disciplines, including architecture. Becker and Becker is a programming organization. It exists professionally primarily to do programming for building owners, architects or anyone else.

The function of this firm in their own words is,"to bring to bear a sum of experience in advanced researching techniques that relate to the human occupancy of space, by a space expert who is an informed, analytical researchist, with a thorough knowledge of design possibilities and limitations. He evaluates all pertinent facts and figures as to their relative worth, collates them into precise needs, and integrates them into a firm, coherent, and meaningful program with an incontrovertible basis in fact."

A pretty tall order if they can do it.

Becker and Becker state their own goals this way:

1) Create optimum Working Environment
2) Organize work and traffic flow
3) Establish effective communications
4) Achieve maximum space utilization
5) Foster good personnel relations, and
6) Provide for orderly growth.

Their approach is achieving these goals is through the following steps:

1) Confer with key personnel

2) Review organizational data

3) Develop personnel projections

4) Analyze personnel traffic

5) Review adjacency requiremnts

6) Establish work flow patterns

7) Inspect presently occupied space

8) Analyze individual work stations

9) Determine furniture and equipment needs

10) Develop work space standards

11) Establish Conference requirements, and

12) Determine shared facilities

Just these simple, clear statements begin to provide some indication of this firm's well organized approach.

While they are to some degree in competition with architects, and feel they can most always do a better job of programming than architects, they prefer to work with architects and do not try to usurp any of the architect's traditional design and decision-making responsibilities. They have worked with some of the top architects on some of the largest projects in the country.

One of the most interesting of Becker and Becker's jobs was the Boston Municipal Building Competition. They were responsible for developing the program on which the competition was based. After the winning design was selected and published, there appeared to be some reluctance

on the part of Boston citizens to accept the "way-out" design. Consequently, Becker and Becker was brought in to analyze the design to see whether it satisfied the program. Fortunately, the design did work very well on the basis of Becker and Becker's analysis and the opponents of the design were apparently pacified. The building is now under construction.

A couple more brief examples of emerging programming techniques and I'll quit.

The architectural firm of Nolen and Swinburne in Philadelphia is in the process of rebuilding their programming procedures through the use of a questionnaire which they hope to pass out to the users of the buildings they have designed. If the program was properly prepared and the building well conceived, the users should reflect this. If something is wrong the Nolen-Swinburne questionnaire should show it.

Nolen-Swinburne also uses a CPM (critical patch schedule) for their entire office operation starting in the programming phase with an economic program, a land program, an environmental program, a human program and a feasibility study. This leads into the second phase consisting of six parallel studies on:

1) Human and building type research
2) Operations Program
3) Spatial Program
4) Building philosophy
5) Site philosophy, and
6) Group dynamics philosophy.

It's interesting to note the prolifiration of different terms used by these different firms.

The firm of Rhone and Ireldale in Vancouver have also developed a CPM for their programming in much more detail and again using different terms. I think the CPM is worth reviewing: they start with,

1) the client-architect agreement
2) the assigmment of the project manager
3) the selection of consultants, and
4) the development of a design budget control diagram, and CPM network.

Before they can proceed further they have to complete step #5, preliminary discussions on site, costs, and so forth.

Step #6 is the topography survey and soil report:

7)site analysis traffic study
8) climatic criteria, and
9) survey of exisiting conditions

At this point they can establish the written statement of direction, and then proceed to all of the next 19 points.

10) report on company image
12) report on future plans
13) local code requirements
14) code extract and by-law envelope criteria
15) local environment
16) space analysis
17) flow diagram
18) material criteria

19) outline specifications & costs

20) mechanical criteria

21) mechancial recommendations & costs

There are similar steps for electrical and structural criteria, recommendations and cost:

28) construction planning

At the completion of all these points they can then proceed to:

29) program cost estimate

30) coordination and review

30) program booklet (and finally)

32) client review and approval

Personally, I am impressed with the straightforwardness of their CPM. I hope soon to learn more aobut it.

In England there has been a great deal of work in programming including the development of the "activity data analysis" method, which has had wide spread use. As usual the British are way ahead of us in some respects--perhaps too mechanical in some other respects.

Jane Hough of the NIMH staff has developed a really wonderful guide to programming for use by NIMH. Other government agencies are following suit.

I have only very briefly touched on some of the highlights of the mass of information I've collected over the past nine months. I have talked about only a few of the people who are doing some noteworthy things in programming. There are obviously many more points (and people) which

could be discussed, and which I will discuss in my final report for the AIA. If all goes well, it will be finished and in print the early part of 1968.

The principal conclusion at which I have arrived in this study is that I am now certain why many of our buildings are not as good as they should be. It's because we are not very good at establishing our goals and aims, and at systemacically making decisions about the future. The programming procedure is the principal weak link in the planning process and we who are responsible for planning the future environment are going to have to increase our competence in this area immediately. I believe we've made long strides in that direction already and I am excited about what the future will bring.